

Binary Serial Data

System feature

Wed, Aug 8, 2007

Normal RS-232 system support assumes that the data transmitted or received is Ascii text, each line of which is terminated by a carriage return (CR) code. This note describes optional support for using the serial port in a Binary mode, for which no such assumption is made; the data is merely a serial stream of binary bytes. The motivation for this work is prompted by a protocol used by the RediStart Micro II, from BenShaw, that is to be used for MuCool helium refrigerators.

In the system code serial port interrupt routine, the normal logic that caches incoming serial data bytes into the `SERIO` system table (circular buffer) ANDs each byte with `0x7F` and looks for a CR code that marks the end of a line of text. In addition, after 250 characters without seeing a CR, it inserts one in place of the most recent byte received. In Binary mode, we need to eliminate this logic. To do so, have the serial interrupt routine, for a serial port operated in Binary mode, invoke a new `SERIOBIN` routine instead of the usual `SERIORCV` routine. The `SERIOBIN` routine can be much shorter and simpler. Each byte that comes in is added into `SERIO`. It increments the `CHARCNT` field in the `SERIO` queue header, and whenever that byte field overflows, it increments the `CRCOUNT` field. These two bytes together could be looked at diagnostically as a 16-bit character counter.

To mark the use of the serial port in Binary mode, we need to set a bit in nonvolatile memory. We can use the upper byte of the word count field in the `PAGEM` table, the first `0x80` bytes of which are used for various system needs for flags. The most significant bit, when set, means Binary mode. This is the word count at `PAGEM+0x16`, which precedes the two-byte register/data bytes to be passed to the ECC serial interface chip during system initialization of the 68K-based IRMs. The usual value seen there is `0x0014`. To denote Binary mode, it should be set to `0x8014`. The code that processes serial port initialization will ignore the hi byte of this word, taking only the low byte to specify the number of words to be processed that follow. The current allowed range for this value is `0x06–0x1F`.

The `SERIO` table is structured as a typical circular buffer implemented for the front end system. There is one `IN` ptr that advances when data is added to the queue, and there are two ptrs, `OUT1` and `OUT2` that advance in sequence toward `IN`. The `Serial` task normally advances `OUT1` when it is made to run after the interrupt routine noticed a CR. This serves to fill `SERIO` for a user of `OUT2` with one or more complete lines of text. The `OUT2` user is normally the read-type routine that fulfills a data request.

To fit in with this normal approach, but do it in the Binary mode, the new `SERIOBIN` routine can advance `IN` and `OUT1` together. (In this case, the `Serial` task will not be made to run.) A request for serial port data can thus work as it is now, but in this new scheme, all data bytes in the `SERIO` will be available for fulfilling the request.